Pāṇinian Syntactico-Semantic Relation Labels

Amba Kulkarni
Department of Sanskrit Studies,
University of Hyderabad
India
apksh@uohyd.ernet.in

Dipti Misra Sharma
Language Technologies Research Center,
IIIT-Hyderabad
India dipti@iiit.ac.in

Abstract

We present in this paper a list of dependency relations based on Pāṇini's grammar for Sanskrit. The important feature of this list is that most of the relations represent well defined semantics that can be extracted from the surface string without any extra-linguistic information.

1 Introduction

In the last two decades the researchers in the Natural Language Processing (NLP) community have recognised the importance of dependency parsing. For English, several parsers producing dependency style output were developed. In the initial stages, there was no consensus among the dependency parser developers on the number of dependency relations and their names. The link parser (Sleator and Temperley, 1993) used 106 relations, while Minipar (Lin, 1998) which was based on Chomsky's minimalism and produced dependency parse used only 59 dependency relations. de Marneffe et al. (2006) modified the dependency relations proposed by Carroll et al. (1999) and King et al. (2003). These relations, known as Stanford Dependencies, were originally developed for English. They proposed a universal taxonomy with a total of 42 relations, which are supported across many languages. This set of relations then was adapted for several other languages. With the development of parsers for several languages, a need was felt to arrive at a single coherent standard, and this led to the development of universal dependencies that can be used for developing cross-linguistically consistent treebanks, that can facilitate multilingual parser development (Nivre, 2015; Nivre et al., 2016). All these various lists of relations mentioned above are syntactic in nature. Several NLP tasks such as database query, robot instructions, information extraction, etc. need semantic representations of sentences. Two major efforts viz. Framenet (Fillmore and Baker, 2000) and Propbank(Kingsbury and Palmer, 2002; Kingsbury and Palmer, 2003) concentrated on the development of semantically tagged lexicon and corpus respectively. The first automatic semantic role labelling system was developed by Gildea and Jurafsky (2002). The major problem with semantic roles is the difficulty involved in coming up with a standard set of roles and formal definitions of thematic roles. As a consequence, PropBank uses verb specific semantic roles as well as generalised semantic roles. Framenet uses semantic roles that are specific to a frame. There are also efforts to transform the syntactic dependency analysis to Logical Form (Reddy et al., 2016) for semantic parsing. There are also efforts to use Abstract Meaning Representation extending the existing relations in Propbank for the development of Semantic databanks (Banarescu et al., 2013).

Given this background, now we highlight some of the salient features of a dependency tagset based on the Pāṇinian grammar framework. Bharati et al. (1991) proposed a computational grammar for processing Indian languages based on the Pāṇinian framework. A dependency tagset based on the Pāṇini's grammar is being used for the development of treebanks for Indian languages (Bharati and Sangal, 1990; Bharati et al., 2002; Rafiya et al., 2008; Chaudhry et al., 2013; Chaudhry and Sharma, 2011). These tagsets are also used for the development of dependency parsers for Indian languages (Tandon and Sharma, 2017). Ramakrishnamacharyulu

(2009) compiled a list of relations used in Indian Grammatical Tradition. A rule-based parser for Sanskrit has been developed using these dependency relations (Kulkarni, 2013; Kulkarni et al., 2010; Kulkarni and Ramakrishnamacharyulu, 2013; Kulkarni, 2019b). There are efforts to analyse English through the Pāṇinian framework. Bhatt (1993) and Bharati et al. (1997) extend the notion of case suffixes (*vibhakti pratyaya*) to account for the notions of subject and object which have fixed positions in a sentence. Bharati and Kulkarni (2011) argues further that the concept of subject in English is the same as the concept of *abhihita* (expressed), and how by assigning a fixed position for Subject, and thereby doing away with the accusative marker English gains in the economy. Sukhada and Sharma (2016) and Bharati et al. (2015) compare the dependencies based on the concepts from Pāṇinian grammar (PG) with other dependency relations such as Stanford Dependencies, Link grammar parser dependencies etc. and offers an automatic mapping of dependency relations of these parsers to a PG based syntactico-semantic scheme.

In the next section, we provide a brief introduction to the Pāṇinian grammar. In the third section, we provide the salient features of the Pāṇinian dependency relations. In the fourth section, we describe the semantic content of the *kāraka* roles (to be defined in the next section) and conclude that this set of relations encodes the semantic relations of predicate arguments that can be extracted without appealing to the world knowledge.

2 Pāṇinian theory of *kāraka* in brief

Sanskrit assumes a unique status when it comes to the field of linguistic analysis with its more than 2500 year long and still extant grammatical tradition. Sanskrit grammar enjoys a similar status in India as mathematics in the West. Pāṇini's grammar is an important milestone in the Indian grammatical tradition. It is the first grammar for any language which is almost complete and together with the theories of verbal understanding (*śābdabodha*), it provides a complete system for language analysis as well as generation for Sanskrit in particular. Pāṇini's grammar known as *Aṣṭādhyāyī* is in the form of aphorisms (*sūtra*s)[1], arranged in 8 chapters with four sections each. According to Kiparsky (2009), the grammar analyses sentences at a hierarchy of 4 levels of description, which are traversed by 3 mappings in the direction from semantics to phonology.

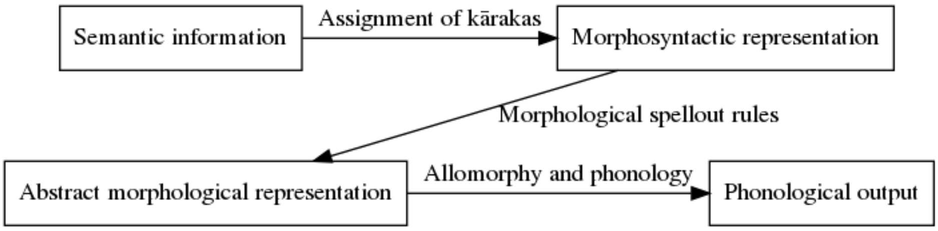
Semantic information
Assignment of kārakas
Morphosyntactic representation
Morphological spellout rules
Abstract morphological representation
Allomorphy and phonology
Phonological output

Figure 1: Levels in the generation process in Pāṇini

The generation starts from the abstract meaning representation and maps it to the surface form incrementally building up from one level to the other. To give an example, the initial semantic representation for the sentence

Skt: Rāmaḥ vanaṁ gacchati
Gloss: Rama {nom.} forest {acc.} go {pr tense, 3p, sg.}
Eng: Ram goes to the forest

may be described as follows:

[1] An aphorism is like a concise formula, the characteristics of which are: minimum number of words, devoid of ambiguity, contains an essence of topic, is universal in nature, without un-meaningful words and without any faults

- there is an activity taking place in the present time,
- there are two participants participating in this activity viz. the doer and the goal.

In the next step, Pāṇini's grammar assigns semantic labels to these various participants. Then the morphological spell out rules assign case suffixes to the participants depending upon their semantic labels, and finally, the phonological rules produce the sentence.

Our main focus of the discussion is on the semantic labels assigned to various participants of the activity. These labels indicate the role (relation) of the participant in the activity, such as *kartṛ*, *karman*, etc. These labels follow directly from the speaker's intention that determines the semantics that would be expressed through the language string. A generic term for such labels is *kāraka*, which literally means "a thing that brings about an action". Pāṇini classifies all these participants into only six categories viz. *kartṛ*, *karman*, *karaṇa*, *sampradāna*, *apādāna* and *adhikaraṇa.* He provides the semantic definitions for them. The definitions go like this.

- The participant which is the most independent to perform the activity is termed as *kartṛ*.[2] (doer of the activity)
- The participant which is the most desired by the *kartṛ* is termed as *karman.*[3] (roughly theme)
- The thing which is most instrumental in bringing the action to accomplishment is called a *karaṇa* (instrument).[4]
- The participant which the agent wishes to reach through the object is termed *sampradāna* (beneficiary).[5]
- The participant which is fixed when there is a movement away is termed as an *apādāna* (source).[6]
- The participant which serves as a locus of an activity is called an *adhikaraṇa* (locus).[7]

These are the general definitions of predicate-argument relations (*kāraka*). Each of these definitions is followed by a list of exceptional cases through which Pāṇini extends the scope of the semantic definitions of the predicate arguments. The extensions are of two types:

- where the associated semantics is totally different from normal expectations and is due to the frozen usages. For example, the verb *sthā* (to stand) takes locus as one argument. But, when this verb is prefixed with *adhi* (the meaning of the verb *adhi-sthā* also has a shade of meaning as 'to govern', in addition 'to stand over', 'to inhabit', etc. by a special rule[8]) the locus gets a *karman* label, as in *saḥ grāmam adhitiṣṭhati* (He inhabits / governs the village). Thus *grāma* (village), here, is not a locus but a theme. Pāṇini lists this rule especially because one may fail to notice this shift in the role when the verbal root has a prefix.
- where the extension to the semantics is not obvious to a layman. In such situations, he lists down special cases making the extension clear and obvious. Such an extension is semantic in nature and is not an idiosyncrasy of Sanskrit. For example, Pāṇini defines the source (*apādānam*) as the participant which is fixed when there is a movement away from it. Thus in *vṛkṣāt parṇam patati* 'The leaf falls from the tree', the tree (*vṛkṣa*) is assigned a role of source (*apādāna*). In the case of a sentence 'The boy fell down from a running horse', the horse is considered to be a source for the action of 'falling down', since the horse, though is running, is stationary relative to the action of falling. He, then, extends this definition to the cases which deal with mental separation and includes verbs such as *bhī* (afraid of) under the purview of this definition. With this, in the sentence, *John is afraid of a lion*, the *lion* gets the source (*apādāna*) role, since John, being afraid of a lion, experiences a mental

[2] *svatantraḥ kartā* (1.4.54) The number in the brackets refer to the chapter.section.*sūtra*

[3] *kartturīpsītatamaṁ karma* (1.4.49)

[4] *sādhakatamaṁ karaṇaṁ* (1.4.42)

[5] *karmaṇā yamabhipraiti sa sampradānaṁ* (1.4.32)

[6] *dhruvam apāye apādānam* (1.4.24)

[7] *ādhāro'dhikaraṇaṁ* (1.4.45)

[8] *adhiśīṅsthāsām karma (1.4.46)* 'in the case of verbal roots *sthā*, *śīṅ* and *as* when prefixed by *adhi* the locus gets a *karman* label

separation from it even when he just thinks of it. Since this extension may not be obvious, Pāṇini provides special aphorisms listing this and all such extensions.

3 Pāṇinian dependency relations for automatic processing

Apart from the predicate-argument relations, Pāṇini also mentions other relations between words such as cause (*hetu*), purpose (*prayojana*), precedence (*pūrvakāla*), etc. without providing any formal definitions for them, and thus implying they carry the same semantics as per their normal language usage. Works, in ancient Indian literature, dealing with grammar (*Vyākaraṇa*), logic (*Nyāya*), and discourse analysis (*Mīmāṁsā*), and especially the texts dealing with the theories of verbal cognition provide a fine-grain classification of such relations.

3.1 Granularity

A list of such relations for Sanskrit was compiled by Ramakrishnamacharyulu (2009). The consortium working on Sanskrit-Hindi Machine Translation adapted a subset of relations from this list for the computational analysis of Sanskrit.[9] It was also noticed that the granularity involved in this collection was too fine for mechanical processing (Kulkarni and Ramakrishnamacharyulu, 2013), and accordingly, a suitable subset was selected that could provide analysis with high accuracy (see Appendix A). The core dependency relations for different modern Indian languages and Sanskrit is common. However, there are a few language specific variations.

3.2 Salient features

Pāṇinian dependency relations have the following features.

- The relations are binary.
- All relations are between words denoting concepts.
- Underspecified relations are provided to handle the complexity in processing.
- Most of the relation names are the same as found in the Pāṇinian tradition. A few new relations, which were not found in Pāṇinian grammar, are added. These correspond to certain accompanying terms (*upapada*) that govern the case markers of the accompanying word. Pāṇini does not discuss the semantics of such relations. Kulkarni (2019a) provides the semantics associated with such relations and thereby elevating the status of such relations from morpho-syntactic to semantic level.
- These dependency relations are found to be suitable for automatic parsing with high accuracy (Kulkarni, 2013).
- The labels are also comprehensible by non-grammarians.
- These relations are also found to be appropriate for both parsing as well as generation (Kulkarni, 2019a).

4 Semantic content

Based on the semantic content, the Pāṇinian dependency relations may be classified into two categories: purely syntactic and purely semantic. We discuss each of them below.

- Purely syntactic
  These tags do not assign any semantic notion to the relation. There are only four such tags.
  - The first one is due to the duplication of a word. There are several meanings associated with the duplication such as pervading, several, successive order, series, distributiveness, repetition, and so on. A Sanskrit word *vīpsā* covers all these meanings. Since in order to decide the exact meaning one needs an access to the extra-linguistic information, we, without analysing this relation further, mark it as *vīpsā*.
  - Another syntactic relation is due to the genitive case marker. The semantic relations associated with this case marker are possession, part and whole relation, kinship rela-

[9] http://sanskrit.uohyd.ac.in/scl/GOLD_DATA/Tagging_Guidelines/tag_proposal_July2019.pdf

tions, and so on. Here also, we do not sub-classify them providing the semantic labels, but collectively classify all of them under the syntactic label genitive (*śaṣṭhī*).

  - The pair of arguments arg1 (*anuyogin*) and arg2 (*pratiyogin*) correspond to the two arguments of a binary relation. They do not carry any specific meaning. These relations are used to specify the inter-sentential relations with sentential connectors such as if-then (*yadi-tarhi*), where the then-clause is the first argument and the if-clause is the second argument with the terms if and then being co-indexed.

- Purely semantic
  Barring the above relations, all other relations are purely semantic in nature. The relations between action and its participants referred to as *kāraka*, and other relations such as purpose (*prayojana*), cause (*hetu*), precedence (*pūrvakāla*) are some examples. The semantics associated with the predicate-argument relations, however, deserves some explanation. Due to the limitation of space, we discuss the semantics associated with only one relation viz. *kartṛ*, and its practical significance from computational point of view.

4.1 *Kartṛ* is not a subject

Consider the analysis of the following two sentences, one in active, and the other in passive represented in Figures 2 and 3 below.
(1) Skt: Rāmaḥ pāṭhaṁ paṭhati
Gloss: Rama{nom.} lesson {acc.} read {pr tense 3p sg}
Eng: Rama reads a lesson.
(2) Skt: Rāmeṇa pāṭhaḥ paṭhyate
Gloss: Rama{ins.} lesson {nom.} read {passive pr tense 3p sg}
Eng: The lesson is read by Rama.

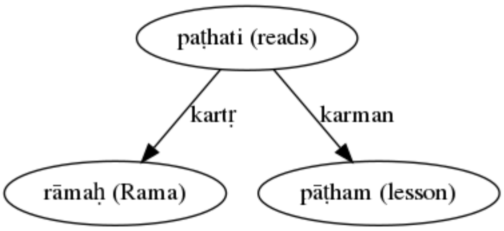

Figure 2: analysis of an active sentence

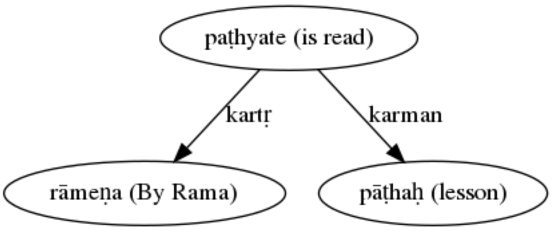

Figure 3: analysis of a passive sentence

We notice that *Rama* which is in the nominative case in the first and in instrument case in the second is marked as *kartṛ* in both the sentences. Special feature of the Pāṇini's grammar is that it does not give two different rules for active and passive, instead handles both by a single rule (Kiparsky, 2009). In other words, there is no transformation rule involved. This brings in uniformity in the analysis of a sentence in the active and passive voice. Now the natural question is, then, is *kartṛ* an agent? And again the answer is No.

4.2 Kartṛ is not an agent

Look at the following three sentences.
1) Skt:*rāmaḥ kuñcikayā tālam udghāṭayati.*
Gloss: Rama{nom.} key{ins.} lock{acc} open{pr tense 3p sg}.
Eng: Rama opens the lock with a key.
In this sentence, *Rama* is a *kartṛ* and an agent, the *key* is an instrument, and the *lock* is the goal. Now consider a situation where somebody is trying to open the lock. He tries with several keys, and finally, with one black key, he could open the lock. In such a situation, he utters,

2) Skt:*śyāmā kuñcikā tālam udghāṭayati.*
Gloss: Black{nom.} key{nom} lock{acc.} open{pr tense 3p sg}.
Eng: The black key opens the lock.
Though thematically, the *key* is still an instrument, according to Pāṇini's grammar, in this sentence it is a *kartṛ*. As a final example, let us consider a situation where somebody is trying to open a lock, and even before inserting the key, the lock gets opened on its own. In such a situation, one may utter 'And then he touches the lock and the lock opens'.
3) Skt:*tālaḥ udghāṭyate.*
Gloss: Lock{nom.} open{pr tense 3p sg}.
Eng: The lock opens.
Here, thematically the lock is a theme. However, according to Pāṇinian analysis, in this sentence, the *lock* is a *kartṛ*. Thus we notice that *kartṛ* in the first sentence is an agent, in the second sentence an instrument and in the third it is the theme. *Kartṛ*, therefore, can be roughly translated as 'doer' which need not be animate.

4.3 What is the semantics associated with the *kartṛ*?

Pāṇini defines *kartṛ*[10] as 'the independent participant in the activity'. An activity typically involves more than one participants. The underlying verb expresses the complex activity which consists of subactivities of each of the participants involved. For example, in the case of opening of a lock, three subactivities are very clearly involved (Bharati et al., 1995) , viz.

1. the insertion of a key by an agent,
2. pressing of the levers of the lock by an instrument (key), and
3. moving of the latch and opening of the lock.

Though in practice, to a large extent all the three subactivities 1 through 3 together constitute the activity 'opening a lock', sometimes the subactivities 2 and 3 together are also referred to as 'opening a lock', as noticed above in the second example, and the activity 3 alone is also referred to as 'opening a lock', as we see in the third sentence. Let us call them open$_1$, open$_2$ and open$_3$, respectively.

Pāṇini draws our attention to the following.

1. The verbal roots are finite in numbers while the conceptual space they cover is infinite. In spite of this, the ambiguity resulting due to the overloading can be resolved from the substantive playing the role of *kartṛ*. Such disambiguation is important in rule-based or knowledge-based Machine Translation systems when the source language and target language map the conceptual space differently. For example, in Hindi open$_1$ and open$_2$ correspond to the verbal root 'khola', while open$_3$ corresponds to the verbal root 'khula'.
2. In order to assign the thematic relations, one has to appeal to the extra-linguistic information.

The greatness of the Pāṇini lies in **"identifying exactly how much information is coded and then giving it a semantic interpretation"** (*sūtras* 1.4.23 - 1.4.55). This level of semantics is the one which is achievable/reachable through the grammar rules and the language string alone. This puts an upper bound on the analysis, making it very clear what is guaranteed by rule-based or knowledge-based analysis and what is not. We can extract only that which is available in a language string 'without any requirement of additional knowledge'.

5 Sanskrit Parser using Pāṇinian Dependencies

A rule-based parser for Sanskrit based on the Indian theories of verbal cognition using the dependency labels provided in Appendix A has been developed which can handle both the prose as well as the verse.[11] For the following verse from the Bhagavadgītā the parser produces the

[10] *svatantraḥ kartā* (1.4.54)

[11] Due to the space constraint, we do not discuss here the performance of this parser. One may refer to (Kulkarni, 2019b) which discusses its performance on the prose.

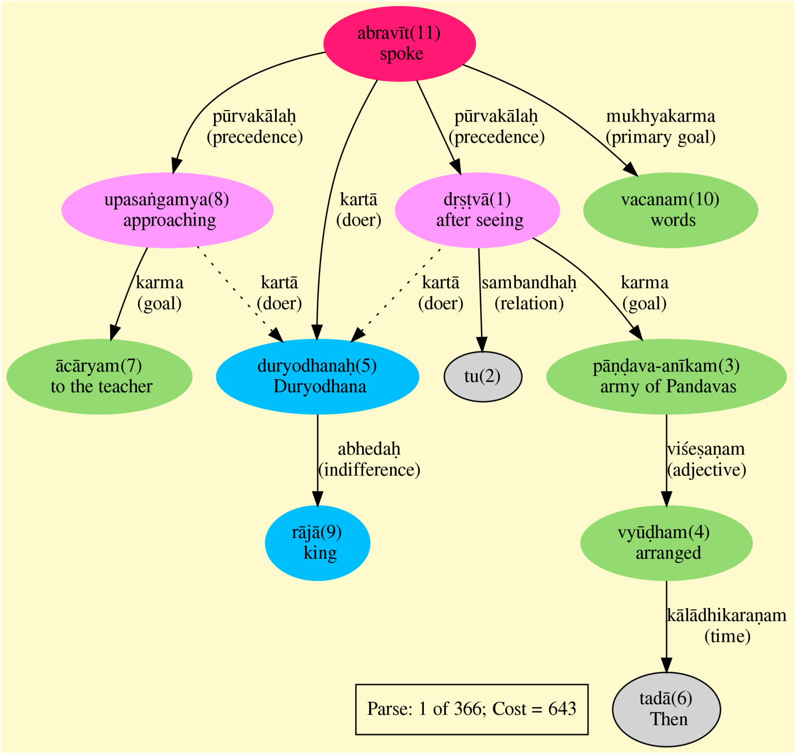

Figure 4: Parsed output of the BhG 1.2 verse

parsed structure as shown in Figure 4. The numbers in the parenthesis indicate the index of the word in the verse. The dotted lines show the shared relations.
Skt: *dṛṣṭvā tu pāṇḍavānīkam vyūḍham duryodhanaḥ tadā* /
ācāryam upasaṅgamya rājā vacanam abravīt // (BhG 1.2)
Gloss: After_seeing[12] the_army_of_the_Pāṇḍavas arranged_in_military_phalanx Duryodhana at_that_time, teacher approached King words spoke
Eng: At that time, after seeing the army of the Pāṇḍavas arranged in military phalanx, King Duryodhana approached (his) teacher and spoke (these) words.

The parser has produced total 366 parses. The first parse is shown here. We note that the parser has gone wrong only in one relation. The sixth word *tadā* (then) should have been connected to the final verb *abravaīt* (spoke). The multiple parses are due to the fact that the parser does not yet have a mechanism to check the mutual compatibility between the word meanings before establishing a relation between them. The current implementation uses this condition only to handle the adjectival relations, where Pāṇini's grammar provides a semantico-syntactic criterian for adjectives, which are otherwise indistinguishable from the substantives morphologically. There are several other cases of ambiguities as well where more than one relation use the same case marker, and the clue is only in the semantics of the word involved. While minimum semantic information such as the classification of the words following the Vaiśeṣka[13] ontology promises better results, the deep learning would complement it further for better results.

6 Conclusion

There are two advantages of using Pāṇinian dependencies. It provides a well-defined semantics that can be extracted purely from the language string. And the same set of relations can be used for both analysis as well as generation. The clear separation of what can be extracted from a language string alone and what can not be helps us plan eclectic use of rule-based and machine

[12] *tu* here is just a filler for metrical purpose
[13] One of the schools of Indian philosophy

learning approaches for developing better parsers.

Acknowledgement

Prof. Vineet Chaitanya deserves special acknowledgement, series of discussions with whom have helped in shaping this manuscript.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Association Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bullgaria. Association for Computational Linguistics.

Akshar Bharati and Amba Kulkarni. 2010. Information coding in a language: Some insights from Paninian grammar. *Dhīmahi, Journal of Chinmaya International Foundation Shodha Sansthan*, I(1):77–91.

Akshar Bharati and Amba Kulkarni. 2011. 'subject' in English is abhihita. In Ashok Aklujkar George Cardona and Hideyo Ogawa, editors, *Studies in Sanskrit Grammars (Proceedings of the Vyakarana Section of the 14th World Sanskrit Conference)*. D.K. Printworld.

Akshar Bharati and Rajeev Sangal. 1990. A karaka based approach to parsing of Indian languages. In *Proceedings of International Conference on Computational Linguistics (Vol. 3)*, Helsinki, Association for Computational Linguistics NY.

Akshar Bharati and Rajeev Sangal. 1993. Parsing free word orderlanguages in the Paninian framework. In *Proceedings of Annual Meetingof Association for Computational Linguistics, Association forComputational Linguistics, New Jersey*, pages 105–111.

Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal. 1991. A computational grammar for Indian languages processing. *Indian Linguistics*, 52, nos 1–4:91–103.

Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal. 1995. *Natural Language Processing: A Paninian Perspective*. Prentice-Hall New Delhi.

Akshar Bharati, Medhavi Bhatia, Vineet Chaitanya, and Rajeev Sangal. 1997. Paninian grammar framework applied to English. *South Asian Langauge Review*.

Akshar Bharati, Rajeev Sangal, Vineet Chaitanya, Amba P Kulkarni, Dipti M Sharma, and KV Ramakrishnamacharyulu. 2002. Anncorra: Building tree-banks in Indian languages. In *Proceedings of Workshop on Asian Language Resources, COLING-2002, Taipei.*

Akshar Bharati, Samar Husain, Dipti M Sharma, and Rajeev Sangal. 2009. Two stage constraint based hybrid approach to free word order language dependency parsing. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT). Paris.*

Akshar Bharati, Sukhada, Dipti M. Sharma, and Soma Paul. 2015. Anusaaraka dependency schema from Paninian perspective. In *Sanskrit and Computational Linguistics, Proceedings of 16th World Sanskrit Conference, Bangkok*. D. K. Publishers.

Rajesh Bhatt. 1993. *Paninian Theory for English*. Ph.D. thesis, Department of CSE, IIT KAnpur. B.Tech. thesis.

J. Bronkhorst. 1979. The role of meanings in pāṇini's grammar. *Indian Linguistics*, 40:146–157.

George Cardona. 2007. On the structure of pāṇṇini's system. *Sanskrit Computational Linguistic*, 1&2:1–31.

John Carroll, Guido Minnen, and Ted Briscoe. 1999. Corpus annotation for parser evaluation. In *Proceedings of the EACL.*

Himani Chaudhry and Dipti M Sharma. 2011. Annotation and issues in building an English dependency treebank. In *Proceedings of 9th international Conference on Natural Language Processing.* Macmillan Publishers, India.

Himani Chaudhry, Himanshu Sharma, and Dipti Misra Sharma. 2013. Divergences in English-Hindi parallel dependency treebanks. In *Procedings of the second international conference on Dependency Linguistics*.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*.

Marie-Catherine de Marneffe, Miriam Connor, Natalia Silveira, Samuel R. Bowman, Timothy Dozat, and Christopher D. Manning. 2013. More constructions, more genres: extending Stanford dependencies.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of LREC*.

Charles J. Fillmore and Collin F. Baker. 2000. Framenet: Frame semantics meets thecorpus. In *Poster presentation, 74th Annual Meeting of the Linguistic Society of America*.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

S.D. Joshi and J.A.F. Roodbergen. 1980. Patañjali's vyākaraṅa-mahābhāṣya, kārakāhnika.

S. D. Joshi. 2009. Background of the aṣṭādhyāayī. pages 1–5.

Tracy H. King, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald Kaplan. 2003. The parc 700 dependency bank. In *4th International Workshop on Linguistically Interpreted Corpora*.

Paul Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *Third International Conference on Language Resources and Evaluation*.

Paul Kingsbury and Martha Palmer. 2003. Propbank: The next level of treebank. In *The Second Workshop on Treebanks and Linguistic Theories*.

Paul Kiparsky. 2009. On the architecture of pāṇini's grammar. pages 32–94.

Amba Kulkarni and K. V. Ramakrishnamacharyulu. 2013. Parsing Sanskrit texts: Some relation specific issues. In Malhar Kulkarni, editor, *Proceedings of the 5th International Sanskrit Computational Linguistics Symposium*. D. K. Printworld(P) Ltd.

Amba Kulkarni, Sheetal Pokar, and Devanand Shukl. 2010. Designing a Constraint Based Parser for Sanskrit. In G N Jha, editor, *Fourth International Sanskrit Computational Linguistics Symposium*, pages 70–90. Springer-Verlag, LNAI 6465.

Amba Kulkarni. 2013. A deterministic dependency parser with dynamic programming for Sanskrit. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, Prague, Czech Republic, August. Charles University in Prague.

Amba Kulkarni. 2019a. Appropriate dependency tagset for Sanskrit analysis and generation. In *Proceedings of Sanskrit in China International Conference 2019: Sanskrit on Paths*. forthcoming.

Amba Kulkarni. 2019b. *Sanskrit Parsing based on the theories of śābdabodha*. D K Print World and IIAS Shimla.

Dekang Lin. 1998. Dependency-based evaluation of MINIPAR. In *Workshop on the evaluation of Parsing Systems*. Granada, Spain.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of LREC*.

Joakim Nivre. 2015. Towards a universal grammar for natural language processing. In *Computational Linguistics and Intelligent Text Processing*.

Sanjeev Panchal and Amba Kulkarni.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James Martin, and Daniel Jurafsky. 2005. Semantic role labeling using different syntactic views. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 581–588.

Begum Rafiya, Samar Husain, Arun Dhwaj, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2008. Dependency annotation scheme for Indian languages. In *Proceedings of The Third International Joint Conference on NaturalLanguage Processing (IJCNLP). Hyderabad, India.*

K V Ramakrishnamacharyulu. 2009. Annotating Sanskrit texts based on Śābdabodha systems. In Amba Kulkarni and Gérard Huet, editors, *Proceedings Third International Sanskrit Computational Linguistics Symposium*, pages 26–39, Hyderabad, India. Springer-Verlag LNAI 5406.

Siva Reddy, Oscar Täckstrom̈, Micheal Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

Daniel D Sleator and Davy Temperley. 1993. Parsing English with a link grammar. In *Third international workshop on Parsing Technologies.*

Sukhada and Dipti M Sharma. 2016. Analyzing English phrases from Paninian perspective. In *17th International Conference on Intelligent Text Processing and Computational Linguistics.*

Juhi Tandon and Dipti Misra Sharma. 2017. Unity in diversity: A unified parsing strategy for major indian languages. In *Proceedings of the fourth International Conference on Dependency Linguistics, Pisa, Italy*, pages 255–265.

Appendices

A Tagset of Dependency Relations

- sambandhaḥ (relation)
  - **Kāraka sambandhāḥ** (Predicate-argument relations)
  - kartā (roughly agent)
    - prayojaka-kartā (causer)
    - prayojya-kartā (causee)
  - karma (goal)
    - mukhya-karma (primary goal)
    - gauṇa-karma (secondary goal)
    - vākya-karma (sentential argument)
  - karaṇaṁ (instrument)
  - sampradānaṁ (beneficiary)
  - apādānaṁ (source)
  - adhikaraṇaṁ (locus)
    - kāla-adhikaraṇaṁ (time)
    - deśa-adhikaraṇaṁ (space)
    - viṣaya-adhikaraṇaṁ (subject as a locus)
  - **Kāraketara sambandhāḥ** (Relations other than arguments of predicate)
    - **kriyā-kriyā-sambandhāḥ** (verb-verb relations)
      - pūrva-kālaḥ (precedence)
      - Vartamāna-samāna-kālaḥ (present participial)
      - Bhaviṣyat-samāna-kālaḥ (future participial)
      - Bhāvalakṣaṇa-pūrva-kālaḥ (simulateity in past)
      - Bhāvalakṣaṇa-vartamāna-samāna-kālaḥ (simulateity in present)
      - Bhāvalakṣaṇa-anantara-kālaḥ (simulateity in future)
      - Sahāyaka-kriyā (auxiliary-verb)
    - **kriyā-sambandhāḥ**
      - sambodhyaḥ (vocative)
      - hetuḥ (cause)
      - prayojanam (purpose)

        - kartṛ-samānādhikaraṇam (predicative adjective)
        - karma-samānādhikaraṇam
        - kriyāviśeṣaṇam (manner adverb)
        - pratiṣedhaḥ (negation)
    - **Nāma-Nāma-sambandhāḥ**
        - śaṣṭhī-sambandhaḥ (genitive)
        - aṅgavikāraḥ (deformity)
        - vīpsā (reduplication)
        - viśeṣaṇam (adjective)
        - sambodhana-sūcakam (vocative marker)
        - vibhaktam
        - avadhiḥ (interval)
        - abhedhaḥ (indifference)
        - nirdhāraṇam
        - atyanta-saṁyogaḥ
        - apavarga-sambandhaḥ
- **Upapada-sambandhāḥ**
    - svāmī (possessor)
    - saha-arthaḥ (association)
    - vinā-arthaḥ (dis-association)
    - point of reference
    - point of comparison
- **Inter-sentential relations**
    - anuyogī (arg1)
    - pratiyogī (arg2)
    - nitya-sambandhaḥ (co-reference)
- **Conjuncts and disjuncts**
    - samuccitaḥ (conjunction)
    - samuccaya-dyotakaḥ (conjunctive marker)
    - anyataraḥ (disjunction)
    - anyatara-dyotakaḥ (disjunctive marker)

Note: The bold entries are the headings and do not indicate relation labels